

WEST Search History

DATE: Sunday, January 25, 2004

Hide?	Set Name	Query	Hit Count
		<i>DB=USPT; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L19	l13 and l18	4
<input type="checkbox"/>	L18	719/311-332[ccls]	2286
<input type="checkbox"/>	L17	l13 and L16	0
<input type="checkbox"/>	L16	719/318[ccls]	190
<input type="checkbox"/>	L15	(object\$ near4 (self adj (describ\$ or contain\$))) same ((defer\$ or delay\$ or laten\$) near4 (rebuild\$ or reconstruct\$ or build\$ or construct\$))	2
<input type="checkbox"/>	L14	L13 and l4	2
<input type="checkbox"/>	L13	(object\$ near4 (self adj (describ\$ or contain\$))) near12 (rebuild\$ or reconstruct\$ or build\$ or construct\$)	190
		<i>DB=EPAB,DWPI; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L12	(object\$ near4 (self adj (describ\$ or contain\$))) near12 (rebuild\$ or reconstruct\$ or build\$ or construct\$)	2
<input type="checkbox"/>	L11	(rpc or (remot\$ adj procedur\$ adj call\$)) and ((delay\$ or defer\$ or laten\$) near4 (rebuild\$ or reconstruct\$ or build\$ or construct\$))	1
<input type="checkbox"/>	L10	(rpc or (remot\$ adj procedur\$ adj call\$)) same (object\$ near4 (rebuild\$ or reconstruct\$ or build\$ or construct\$))	2282
		<i>DB=PGPB; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L9	(rpc or (remot\$ adj procedur\$ adj call\$)) same (object\$ near4 (rebuild\$ or reconstruct\$ or build\$ or construct\$))	13
<input type="checkbox"/>	L8	719/330[ccls]	85
<input type="checkbox"/>	L7	(rpc or (remot\$ adj procedur\$ adj call\$)) same ((delay\$ or defer\$ or laten\$) near4 (rebuild\$ or reconstruct\$ or build\$ or construct\$))	2
		<i>DB=USPT; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L6	(rpc or (remot\$ adj procedur\$ adj call\$)) same ((delay\$ or defer\$ or laten\$) near4 (rebuild\$ or reconstruct\$ or build\$ or construct\$))	7
<input type="checkbox"/>	L5	(rpc or (remot\$ adj procedur\$ adj call\$)) same (object near4 (rebuild\$ or reconstruct\$ or build\$ or construct\$))	65
<input type="checkbox"/>	L4	719/330[ccls]	184
<input type="checkbox"/>	L3	709/330[ccls]	0
<input type="checkbox"/>	L2	(rpc or (remot\$ adj procedur\$ adj call\$)) same ((defer\$ or delay\$ or laten\$) near4 (rebuild\$ or reconstruct\$))	3
<input type="checkbox"/>	L1	(rpc or (remot\$ adj procedur\$ adj call\$)) near12 (rebuild\$ or reconstruct\$)	13

END OF SEARCH HISTORY



US005557798A

United States Patent [19]

Skeen et al.

[11] **Patent Number:** 5,557,798[45] **Date of Patent:** *Sep. 17, 1996

[54] **APPARATUS AND METHOD FOR PROVIDING DECOUPLING OF DATA EXCHANGE DETAILS FOR PROVIDING HIGH PERFORMANCE COMMUNICATION BETWEEN SOFTWARE PROCESSES**

[75] **Inventors:** Marion D. Skeen, Palo Alto; Mark Bowles, Woodside, both of Calif.

[73] **Assignee:** Tibco, Inc., Palo Alto, Calif.

[*] **Notice:** The term of this patent shall not extend beyond the expiration date of Pat. No. 5,257,369.

[21] **Appl. No.:** 632,551

[22] **Filed:** Dec. 21, 1990

Related U.S. Application Data

[63] Continuation-in-part of Ser. No. 601,117, Oct. 22, 1990, Pat. No. 5,257,369, which is a continuation-in-part of Ser. No. 386,584, Jul. 27, 1989, Pat. No. 5,187,787.

[51] **Int. Cl.⁶** G06F 15/16; G06F 13/00

[52] **U.S. CL** 395/650; 364/280; 364/284; 364/284.3; 364/281.3; 364/DIG. 1

[58] **Field of Search** 395/650, 700

References Cited**U.S. PATENT DOCUMENTS**

4,463,093	12/1982	Davis et al.	364/200
4,688,170	8/1987	Waite et al.	364/200
4,718,005	1/1988	Feigenbaum et al.	364/200
4,751,635	6/1988	Kret	
4,815,030	3/1989	Cross et al.	364/900
4,815,988	3/1989	Trottier et al.	364/200
4,823,122	4/1989	Mann et al.	340/825.28
4,851,988	7/1989	Trottier et al.	364/200

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

41417/89	9/1989	Australia	G06F 15/16
41416/89	9/1989	Australia	G06F 13/42
4141789	12/1989	Australia	G06F 15/16

4141689	12/1989	Australia	G06F 13/42
79455/91	6/1991	Australia	G06F 15/16
0108233	5/1984	European Pat. Off.	G06F 15/16
0130375	1/1985	European Pat. Off.	G06F 15/40
0167725	1/1986	European Pat. Off.	G06F 5/00
0216535	8/1986	European Pat. Off.	G06F 15/16
0258867	3/1988	European Pat. Off.	G06F 9/46
0387462	9/1990	European Pat. Off.	G06G 15/21
57-92954	of 1980	Japan	
63-50140	of 1986	Japan	
63-214045	of 1987	Japan	
63-174159	of 1987	Japan	
2191069	12/1987	United Kingdom	G09G 1/00
2205018	11/1988	United Kingdom	G06F 15/40

OTHER PUBLICATIONS

ISIS and the Meta Project; K. Birman and K. Marzullo; published in Sun Technology, Summer 1989.

News Need Not be Slow; G. Collyer and H. Spencer; published in Winter 1987 USENIX Technical Conference; Winter 1987.

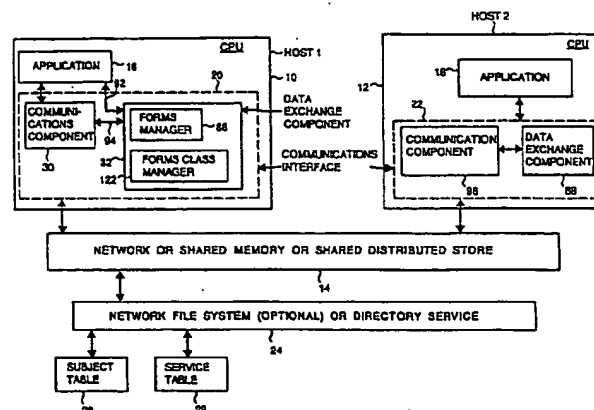
(List continued on next page.)

Primary Examiner—Kevin A. Kriess

Attorney, Agent, or Firm—Ronald Fish; Falk, Vestal & Fish

[57] ABSTRACT

A communication interface for decoupling one software application from another software application such communications between applications are facilitated and applications may be developed in modularized fashion. The communication interface is comprised of two libraries of programs. One library manages self-describing forms which contain actual data to be exchanged as well as type information regarding data format and class definition that contain semantic information. Another library manages communications and includes a subject mapper to receive subscription requests regarding a particular subject and map them to particular communication disciplines and to particular services supplying this information. A number of communication disciplines also cooperate with the subject mapper or directly with client applications to manage communications with various other applications using the communication protocols used by those other applications.

56 Claims, 20 Drawing Sheets

[First Hit](#) [Fwd Refs](#)**End of Result Set**

Generate Collection

Print

L19: Entry 4 of 4

File: USPT

Sep 17, 1996

DOCUMENT-IDENTIFIER: US 5557798 A

TITLE: Apparatus and method for providing decoupling of data exchange details for providing high performance communication between software processes

Current US Cross Reference Classification (3):719/328

CLAIMS:

17. The process of claim 14 wherein data messages transmitted between said data producing processes and said data consuming processes, hereafter referred to as processes, are transmitted as self describing data objects, wherein each self describing data object is comprised of one or more fields each of which is either a primitive class form which stores data or a constructed class form which is comprised of other fields which themselves may be primitive or constructed class forms, each said constructed class form belonging to a class which has a corresponding class definition, said self describing data objects being organized into classes defined by class definitions, each class definition comprising a list of the fields by name and data representation type which are common to all self describing data objects of that class, each self describing data object including both data format information and actual data or field values for each said field, and further comprising the steps of:

automatically converting any self describing data objects to be transmitted from one process to another from the format of the transmitting process to the format necessary for transmission across said data path prior to transmission thereof, and then transmitting said self describing data object through said data path; and

automatically converting any self describing data objects received after transmission through said data path which are bound for either a data consuming process or a data producing process, from the format used to transmit data across said data path to the format used by said receiving process.

19. The process of claim 14 wherein data transmitted between said data producing processes and said data consuming processes is transmitted as self describing data objects, each self describing data object comprised of one or more fields each of which is either a primitive class form which stores data or a constructed class form which is comprised of other fields which themselves may be primitive or constructed class forms, each said constructed class form belonging to a class which has a corresponding class definition, said self describing data objects being organized into classes defined by class definitions, each class definition comprising a list of the fields by name and data representation type which are common to all self describing data objects of that class, each self describing data object including both data format information and actual data or field values for each said field, for providing the capability for a data consuming or data producing process to obtain data from a particular field of a particular self describing data object generated by another process, comprising the steps of:



US006212575B1

(12) **United States Patent**
Cleron et al.

(10) **Patent No.: US 6,212,575 B1**
 (45) **Date of Patent: *Apr. 3, 2001**

(54) **EXTENSIBLE, REPLACEABLE NETWORK COMPONENT SYSTEM**

2 242 293 1/1990 (GB) .

OTHER PUBLICATIONS

- (75) Inventors: **Michael A. Cleron; Stephen Fisher**, both of Menlo Park; **Timo Bruck**, Mountain View, all of CA (US)
- (73) Assignee: **Apple Computer, Inc.**, Cupertino, CA (US)
- (*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

Potel et al; The Architecture of the Taligent System; Dr. Dobbs Journal on CD-ROM, SP 94.*
 Rush, Jeff; OpenDoc; Dr. Dobbs's Journal on CD-ROM, SP 94.*
 Piersol, Kurt; A Close-Up of OpenDoc; AIXpert, Jun. 1994.*
 Schmidt et al; "An object-oriented framework for developing network server daemons", C++ World Conference, pp. 1-15, 10/93.*
 "Leveraging object-oriented frameworks", Taligent white paper, 1993.*
 Andert, Glerk; "Object-Frameworks in the Taligent OS", IEEE electronic Library, pp.112-121, 1994.*
 Helm et al, "Integrating information retrieval and domain specific approaches for browsing and retrieval in object-oriented class libraries", ACM Digital Library, 1991.*

(List continued on next page.)

(21) Appl. No.: **08/435,377**

(22) Filed: **May 5, 1995**

- (51) Int. Cl.⁷ **G06F 9/00; G06F 15/16**
 (52) U.S. Cl. **709/328; 709/200**
 (58) Field of Search **395/700, 200.02, 395/200.03, 200.08, 200.09, 200.32; 709/300-305; 345/356, 557**

(56) **References Cited**

U.S. PATENT DOCUMENTS

- 5,297,249 * 3/1994 Bernstein et al. 345/356
 5,339,430 8/1994 Lundin et al. 395/700
 5,481,666 * 1/1996 Nguyen et al. 345/357
 5,530,852 * 6/1996 Meske, Jr. et al. 395/200.03
 5,537,526 * 7/1996 Anderson .
 5,548,722 * 8/1996 Jalalian .
 5,581,686 * 12/1996 Koppolu et al. 395/340
 5,584,035 * 12/1996 Duggan et al. 395/800
 5,634,129 * 5/1997 Dickinson 709/303
 5,669,005 * 9/1997 Curbow .

FOREIGN PATENT DOCUMENTS

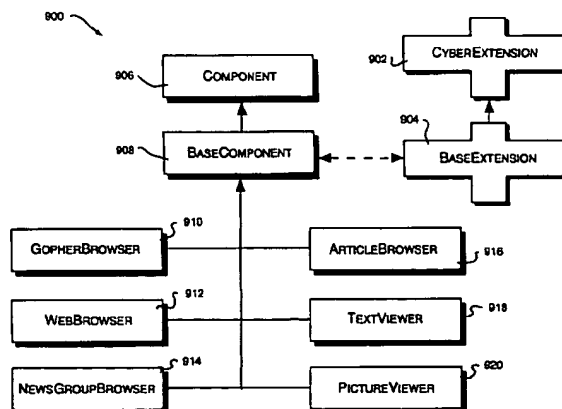
0 631 456 A2 12/1994 (EP) .

Primary Examiner—Majid A. Banankhah
Assistant Examiner—P. G. Caldwell
 (74) *Attorney, Agent, or Firm*—Cesari and McKenna, LLP

(57) **ABSTRACT**

An extensible and replaceable network-oriented component system provides a platform for developing network navigation components that operate on a variety of hardware and software computer systems. These navigation components include key integrating components along with components configured to deliver conventional services directed to computer networks, such as Gopher-specific and Web-specific components. Communication among these components is achieved through novel application programming interfaces (APIs) to facilitate integration with an underlying software component architecture. Such a highly-modular cooperating layered-arrangement between the network component system and the component architecture allows any existing component to be replaced, and allows new components to be added, without affecting operation of the network component system.

13 Claims, 8 Drawing Sheets



[First Hit](#) [Fwd Refs](#)

Generate Collection

Print

L19: Entry 3 of 4

File: USPT

Apr 3, 2001

DOCUMENT-IDENTIFIER: US 6212575 B1

TITLE: Extensible, replaceable network component system

Detailed Description Text (45):

FIG. 7 illustrates a simplified class hierarchy diagram 700 of the base class CyberItem 702 used to construct the network component object 602. In general, CyberItem is an abstraction that may represent resources available at any location accessible from the client 200. However, in accordance with the illustrative embodiment, a CyberItem is preferably a small, self-contained object that represents a resource, such as a service, available on the Internet and subclasses of the CyberItem base class are used to construct various network component objects configured to provide such services for the novel network-oriented component system.

Current US Original Classification (1):719/328